## CLAIMS

**What is claimed is:**

1.      A method in a data processing system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the method comprising the steps of:

       displaying a graphical representation of the dependent node;

       displaying a graphical representation of the defining node;

       receiving an indication to identify a dependency between the dependent node and the defining node; and

       in response to receiving the indication to identify the dependency, representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node.

2.      The method of claim 1, wherein the plurality of dependencies is represented as a single link.

3.      The method of claim 1, wherein each node comprises an element.

4.      The method of claim 3, wherein one of the plurality of dependencies comprises a use of the defining node element by the dependent node element.

5.      The method of claim 3, wherein one of the plurality of dependencies comprises a declaration of the defining node element by the dependent node element.

6.      The method of claim 3, wherein one of the plurality of dependencies comprises a call to a method of the defining node element by the dependent node element.

7.      The method of claim 3, wherein one of the plurality of dependencies comprises a local variable definition using the defining node element in a method of the dependent node element.

8.      The method of claim 3, wherein the dependent node element comprises a class.

9.      The method of claim 3, wherein the dependent node element comprises an interface.

10. The method of claim 3, wherein the defining node element comprises a class.

11. The method of claim 3, wherein the defining node element comprises an interface.

12. The method of claim 1, wherein the dependent node comprises a class.

13. The method of claim 1, wherein the dependent node comprises an interface.

14. The method of claim 1, wherein the dependent node comprises a package.

15. The method of claim 14, wherein the package comprises a plurality of elements.

16. The method of claim 15, wherein one of the plurality of elements comprises a class.

17. The method of claim 15, wherein one of the plurality of elements comprises an interface.

18. The method of claim 1, wherein the defining node comprises a class.

19. The method of claim 1, wherein the defining node comprises an interface.

20. The method of claim 1, wherein the defining node comprises a package.

21. The method of claim 20, wherein the package comprises a plurality of elements.

22. The method of claim 21, wherein one of the plurality of elements comprises a class.

23. The method of claim 21, wherein one of the plurality of elements comprises an interface.

24.      A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

displaying a graphical representation of the plurality of nodes;

determining whether the code corresponding to a first of the plurality of nodes contains a first use of a second of the plurality of nodes; and

when it is determined that the code corresponding to the first node contains the first use of the second node,

determining whether the code corresponding to the first node contains a second use of the second node; and

when it is determined that the code corresponding to the first node contains the second use of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

25.      The method of claim 24, wherein the first use comprises a declaration.

26.      The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an attribute declaration that uses the second node.

27.      The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an initializer of an attribute declaration that uses the second node.

28.      The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an argument parameter of a method that uses the second node.

29.      The method of claim 24, wherein the first use comprises a method call.

30.     The method of claim 24, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

31.     The method of claim 24, wherein the first node comprises a class.

32.     The method of claim 24, wherein the first node comprises an interface.

33.     The method of claim 24, wherein the first node comprises a package.

34.     The method of claim 24, wherein the second node comprises a class.

35.     The method of claim 24, wherein the second node comprises an interface.

36.     The method of claim 24, wherein the second node comprises a package.

37.    A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

displaying a graphical representation of the plurality of nodes;

determining whether the code corresponding to a first of the plurality of nodes contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the declaration of the second node,

determining whether the code corresponding to the first node contains another declaration of the second node; and

when it is determined that the code corresponding to the first node contains the other declaration of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

38.    The method of claim 37, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node,

determining whether the code corresponding to the first node contains a call to a method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

39.    The method of claim 37, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node,

determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

40. The method of claim 37, wherein the first node comprises a class.

41. The method of claim 37, wherein the first node comprises an interface.

42. The method of claim 37, wherein the first node comprises a package.

43. The method of claim 37, wherein the second node comprises a class.

5  44. The method of claim 37, wherein the second node comprises an interface.

45. The method of claim 37, wherein the second node comprises a package.

46.     A method in a data processing system having a first node and a second node, each of the nodes having corresponding code, the second node having a method, the method comprising the steps of:

displaying a graphical representation of the first node and the second node;

determining whether the code corresponding to the first node contains a call to the method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node,

determining whether the code corresponding of the first node contains another call to the method of the second node; and

when it is determined that the code corresponding to the first node contains the other call to the method of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

47.     The method of claim 46, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node,

determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

48. The method of claim 46, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node,

   determining whether the code corresponding to the first node contains a declaration of the second node; and

   when it is determined that the code corresponding to the first node contains the declaration of the second node,

   displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

49. The method of claim 46, wherein the first node comprises a class.

50. The method of claim 46, wherein the first node comprises an interface.

51. The method of claim 46, wherein the first node comprises a package.

52. The method of claim 46, wherein the second node comprises a class.

53. The method of claim 46, wherein the second node comprises an interface.

54. The method of claim 46, wherein the second node comprises a package.

55.    A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a first use and a second use of the second node; and

when it is determined that the code corresponding to the first node contains the first use and the second use of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

56.    The method of claim 55, wherein the first use comprises a declaration.

57.    The method of claim 55, wherein the first use comprises a method call.

58.    The method of claim 55, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

59.    The method of claim 55, wherein the first node comprises a class.

60.    The method of claim 55, wherein the first node comprises an interface.

61.    The method of claim 55, wherein the first node comprises a package.

62.    The method of claim 55, wherein the second node comprises a class.

63.    The method of claim 55, wherein the second node comprises an interface.

64.    The method of claim 55, wherein the second node comprises a package.

65. A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

5      determining whether the code corresponding to the first node contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the declaration of the second node,

determining whether the code corresponding to the first node contains another declaration of the second node; and

10         when it is determined that the code corresponding to the first node contains the other declaration of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second

15            node.

66. The method of claim 65, wherein the first node comprises a class.

67. The method of claim 65, wherein the first node comprises an interface.

68. The method of claim 65, wherein the first node comprises a package.

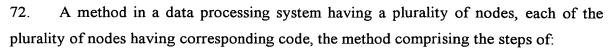69. The method of claim 65, wherein the second node comprises a class.

20   70. The method of claim 65, wherein the second node comprises an interface.

71. The method of claim 65, wherein the second node comprises a package.

72.   A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes, wherein the second node has a method;

determining whether the code corresponding to the first node contains a call to the method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node,

determining whether the code corresponding of the first node contains another call to the method of the second node; and

when it is determined that the code corresponding to the first node contains another call to the method of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

73.   The method of claim 72, wherein the first node comprises a class.

74.   The method of claim 72, wherein the first node comprises an interface.

75.   The method of claim 72, wherein the first node comprises a package.

76.   The method of claim 72, wherein the second node comprises a class.

77.   The method of claim 72, wherein the second node comprises an interface.

78.   The method of claim 72, wherein the second node comprises a package.

79. A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

5 determining whether the code corresponding to the first node contains a use of the second node; and

when it is determined that the code corresponding to the first node contains the use of the second node,

displaying the usage of the second node by the first node.

10 80. The method of claim 79, wherein each node comprises an element.

81. The method of claim 79, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the first node element.

82. The method of claim 81, further comprising the step of displaying the first node
15 with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

83. The method of claim 79, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the second node element.

20 84. The method of claim 83, further comprising the step of displaying the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

85. The method of claim 79, wherein the first node comprises a class.

86. The method of claim 79, wherein the first node comprises an interface.

25 87. The method of claim 79, wherein the first node comprises a package.

88. The method of claim 79, wherein the second node comprises a class.

89. The method of claim 79, wherein the second node comprises an interface.

90. The method of claim 79, wherein the second node comprises a package.

91.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the method comprising the steps of:

5            displaying a graphical representation of the dependent node;

             displaying a graphical representation of the defining node;

             receiving an indication to identify a dependency between the dependent node and the defining node; and

             in response to receiving the indication to identify the dependency, representing

10           the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node.

92.    The computer-readable medium of claim 91, wherein the plurality of dependencies is represented as a single link.

93.    The computer-readable medium of claim 91, wherein each node comprises an

15    element.

94.    The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a use of the defining node element by the dependent node element.

95.    The computer-readable medium of claim 93, wherein one of the plurality of

20    dependencies comprises a declaration of the defining node element by the dependent node element.

96.    The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a call to a method of the defining node element by the dependent node element.

25    97.    The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a local variable definition using the defining node element in a method of the dependent node element.

98.    The computer-readable medium of claim 93, wherein the dependent node element comprises a class.

99.    The computer-readable medium of claim 93, wherein the dependent node element comprises an interface.

5    100.    The computer-readable medium of claim 93, wherein the defining node element comprises a class.

101.    The computer-readable medium of claim 93, wherein the defining node element comprises an interface.

102.    The computer-readable medium of claim 91, wherein the dependent node

10    comprises a class.

103.    The computer-readable medium of claim 91, wherein the dependent node comprises an interface.

104.    The computer-readable medium of claim 91, wherein the dependent node comprises a package.

15    105.    The computer-readable medium of claim 104, wherein the package comprises a plurality of elements.

106.    The computer-readable medium of claim 105, wherein one of the plurality of elements comprises a class.

107.    The computer-readable medium of claim 105, wherein one of the plurality of

20    elements comprises an interface.

108.    The computer-readable medium of claim 91, wherein the defining node comprises a class.

109.    The computer-readable medium of claim 91, wherein the defining node comprises an interface.

110.   The computer-readable medium of claim 91, wherein the defining node comprises a package.

111.   The computer-readable medium of claim 110, wherein the package comprises a plurality of elements.

5   112.   The computer-readable medium of claim 111, wherein one of the plurality of elements comprises a class.

113.   The computer-readable medium of claim 111, wherein one of the plurality of elements comprises an interface.

114.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

5          displaying a graphical representation of the plurality of nodes;

           determining whether the code corresponding to a first of the plurality of nodes contains a first use of a second of the plurality of nodes; and

           when it is determined that the code corresponding to the first node contains the first use of the second node,

10              determining whether the code corresponding to the first node contains a second use of the second node; and

                when it is determined that the code corresponding to the first node contains the second use of the second node,

                    displaying a dependency link between the graphical representation

15                  of the first node and the graphical representation of the second node.

115.    The computer-readable medium of claim 114, wherein the first use comprises a declaration.

116.    The computer-readable medium of claim 115, wherein the step of determining

20   whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an attribute declaration that uses the second node.

117.    The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node

25   comprises the step of searching the code corresponding to the first node for an initializer of an attribute declaration that uses the second node.

118.   The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an argument parameter of a method that uses the second node.

5   119.   The computer-readable medium of claim 114, wherein the first use comprises a method call.

120.   The computer-readable medium of claim 114, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

10   121.   The computer-readable medium of claim 114, wherein the first node comprises a class.

122.   The computer-readable medium of claim 114, wherein the first node comprises an interface.

123.   The computer-readable medium of claim 114, wherein the first node comprises a
15   package.

124.   The computer-readable medium of claim 114, wherein the second node comprises a class.

125.   The computer-readable medium of claim 114, wherein the second node comprises an interface.

20   126.   The computer-readable medium of claim 114, wherein the second node comprises a package.

127.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

5           displaying a graphical representation of the plurality of nodes;

determining whether the code corresponding to a first of the plurality of nodes contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the declaration of the second node,

10              determining whether the code corresponding to the first node contains another declaration of the second node; and

when it is determined that the code corresponding to the first node contains the other declaration of the second node,

displaying a dependency link between the graphical representation

15              of the first node and the graphical representation of the second node.

128.    The computer-readable medium of claim 127, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the method further comprises the steps of:

20          determining whether the code corresponding to the first node contains a call to a method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node,

displaying a dependency link between the graphical representation of the

25              first node and the graphical representation of the second node.

129.     The computer-readable medium of claim 127, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the method further comprises the steps of:

　　　determining whether the code corresponding to the first node comprises a method
having a local variable definition using the second node; and

　　　when it is determined that the code corresponding to the first node comprises a
method having the local variable definition using the second node,

　　　　　displaying a dependency link between the graphical representation of the
first node and the graphical representation of the second node.

130.     The computer-readable medium of claim 127, wherein the first node comprises a class.

131.     The computer-readable medium of claim 127, wherein the first node comprises an interface.

132.     The computer-readable medium of claim 127, wherein the first node comprises a package.

133.     The computer-readable medium of claim 127, wherein the second node comprises a class.

134.     The computer-readable medium of claim 127, wherein the second node comprises an interface.

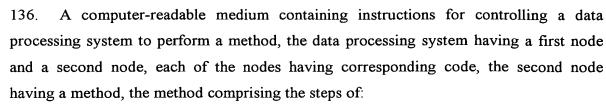135.     The computer-readable medium of claim 127, wherein the second node comprises a package.

136.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a first node and a second node, each of the nodes having corresponding code, the second node having a method, the method comprising the steps of:

5          displaying a graphical representation of the first node and the second node;

determining whether the code corresponding to the first node contains a call to the method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node,

10              determining whether the code corresponding of the first node contains another call to the method of the second node; and

when it is determined that the code corresponding to the first node contains the other call to the method of the second node,

displaying a dependency link between the graphical representation

15              of the first node and the graphical representation of the second node.

137.    The computer-readable medium of claim 136, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the method further comprises the steps of:

20          determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node,

displaying a dependency link between the graphical representation of the

25              first node and the graphical representation of the second node.

138. The computer-readable medium of claim 136, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the method further comprises the steps of:

determining whether the code corresponding to the first node contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the declaration of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

139. The computer-readable medium of claim 136, wherein the first node comprises a class.

140. The computer-readable medium of claim 136, wherein the first node comprises an interface.

141. The computer-readable medium of claim 136, wherein the first node comprises a package.

142. The computer-readable medium of claim 136, wherein the second node comprises a class.

143. The computer-readable medium of claim 136, wherein the second node comprises an interface.

144. The computer-readable medium of claim 136, wherein the second node comprises a package.

145.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

5          receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a first use and a second use of the second node; and

when it is determined that the code corresponding to the first node contains the

10         first use and the second use of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

146.    The computer-readable medium of claim 145, wherein the first use comprises a declaration.

15    147.    The computer-readable medium of claim 145, wherein the first use comprises a method call.

148.    The computer-readable medium of claim 145, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

20    149.    The computer-readable medium of claim 145, wherein the first node comprises a class.

150.    The computer-readable medium of claim 145, wherein the first node comprises an interface.

151.    The computer-readable medium of claim 145, wherein the first node comprises a

25    package.

152.    The computer-readable medium of claim 145, wherein the second node comprises a class.

153.    The computer-readable medium of claim 145, wherein the second node comprises an interface.

154.    The computer-readable medium of claim 145, wherein the second node comprises a package.

155.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

5          receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the

10        declaration of the second node,

determining whether the code corresponding to the first node contains another declaration of the second node; and

when it is determined that the code corresponding to the first node contains the other declaration of the second node,

15                displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

156.    The computer-readable medium of claim 155, wherein the first node comprises a class.

157.    The computer-readable medium of claim 155, wherein the first node comprises

20        an interface.

158.    The computer-readable medium of claim 155, wherein the first node comprises a package.

159.    The computer-readable medium of claim 155, wherein the second node

25        comprises a class.

160.    The computer-readable medium of claim 155, wherein the second node comprises an interface.

161. The computer-readable medium of claim 155, wherein the second node comprises a package.

162. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

5           receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes, wherein the second node has a method;

determining whether the code corresponding to the first node contains a call to the method of the second node; and

10           when it is determined that the code corresponding to the first node contains the call to the method of the second node,

determining whether the code corresponding of the first node contains another call to the method of the second node; and

when it is determined that the code corresponding to the first node

15           contains another call to the method of the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

163. The computer-readable medium of claim 162, wherein the first node comprises a

20   class.

164. The computer-readable medium of claim 162, wherein the first node comprises an interface.

165. The computer-readable medium of claim 162, wherein the first node comprises a package.

25   166. The computer-readable medium of claim 162, wherein the second node comprises a class.

167. The computer-readable medium of claim 162, wherein the second node comprises an interface.

168. The computer-readable medium of claim 162, wherein the second node comprises a package.

169.    A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

5          receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a use of the second node; and

when it is determined that the code corresponding to the first node contains the

10        use of the second node,

displaying the usage of the second node by the first node.

170.    The computer-readable medium of claim 169, wherein each node comprises an element.

171.    The computer-readable medium of claim 169, wherein when it is determined that

15  the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the first node element.

172.    The computer-readable medium of claim 171, wherein the method further comprises the step of displaying the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

20  173.    The computer-readable medium of claim 169, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the second node element.

174.    The computer-readable medium of claim 173, wherein the method further comprises the step of displaying the second node with the second node element and with

25  the usage to visually indicate that the first node contains the usage of the second node.

175.    The computer-readable medium of claim 169, wherein the first node comprises a class.

176.     The computer-readable medium of claim 169, wherein the first node comprises an interface.

177.     The computer-readable medium of claim 169, wherein the first node comprises a package.

5     178.     The computer-readable medium of claim 169, wherein the second node comprises a class.

179.     The computer-readable medium of claim 169, wherein the second node comprises an interface.

180.     The computer-readable medium of claim 169, wherein the second node

10     comprises a package.

181. A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;

a memory device further comprising a program that displays a graphical representation of the plurality of nodes, that determines whether the code corresponding to a first of the plurality of nodes contains a declaration of the second node, and when it is determined that the code corresponding to the first node contains the declaration of the second node, the program determines whether the code corresponding to the first node contains another declaration of the second node, and when it is determined that the code corresponding to the first node contains the other declaration of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and

a processor for running the program.

182. The data processing system of claim 181, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the program determines whether the code corresponding to the first node contains a call to a method of the second node, and when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

183. The data processing system of claim 181, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the program determines whether the code corresponding to the first node comprises a method having a local variable definition using the second node, and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

184.     The data processing system of claim 181, wherein the first node comprises a class.

185.     The data processing system of claim 181, wherein the first node comprises an interface.

5     186.     The data processing system of claim 181, wherein the first node comprises a package.

187.     The data processing system of claim 181, wherein the second node comprises a class.

188.     The data processing system of claim 181, wherein the second node comprises an
10     interface.

189.     The data processing system of claim 181, wherein the second node comprises a package.

190.  A data processing system comprising:

a secondary storage device further comprising a first node and a second node, each of the nodes having corresponding code, the second node having a method;

a memory device further comprising a program that displays a graphical representation of the first node and the second node, that determines whether the code corresponding to the first node contains a call to the method of the second node, and when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program determines whether the code corresponding of the first node contains another call to the method of the second node, and when it is determined that the code corresponding to the first node contains the other call to the method of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and

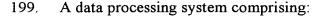a processor for running the program.

191.  The data processing system of claim 190, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the program determines whether the code corresponding to the first node comprises a method having a local variable definition using the second node, and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

192.     The data processing system of claim 190, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the program determines whether the code corresponding to the first node contains a declaration of the second node, and when it is determined that the code corresponding to the first node contains the declaration of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

193.     The data processing system of claim 190, wherein the first node comprises a class.

194.     The data processing system of claim 190, wherein the first node comprises an interface.

195.     The data processing system of claim 190, wherein the first node comprises a package.

196.     The data processing system of claim 190, wherein the second node comprises a class.

197.     The data processing system of claim 190, wherein the second node comprises an interface.

198.     The data processing system of claim 190, wherein the second node comprises a package.

199. A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;

a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that receives an indication of a second of the plurality of nodes, that determines whether the code corresponding to the first node contains a declaration of the second node, and when it is determined that the code corresponding to the first node contains the declaration of the second node, the program determines whether the code corresponding to the first node contains another declaration of the second node, and when it is determined that the code corresponding to the first node contains the other declaration of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and

a processor for running the program.

200. The data processing system of claim 199, wherein the first node comprises a class.

201. The data processing system of claim 199, wherein the first node comprises an interface.

202. The data processing system of claim 199, wherein the first node comprises a package.

203. The data processing system of claim 199, wherein the second node comprises a class.

204. The data processing system of claim 199, wherein the second node comprises an interface.

205. The data processing system of claim 199, wherein the second node comprises a package.

206.    A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;
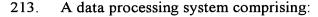
a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that receives an indication of a second of the plurality of nodes, wherein the second node has a method, that determines whether the code corresponding to the first node contains a call to the method of the second node, and when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program determines whether the code corresponding of the first node contains another call to the method of the second node, and when it is determined that the code corresponding to the first node contains another call to the method of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and

a processor for running the program.

207.    The data processing system of claim 206, wherein the first node comprises a class.

208.    The data processing system of claim 206, wherein the first node comprises an interface.

209.    The data processing system of claim 206, wherein the first node comprises a package.

210.    The data processing system of claim 206, wherein the second node comprises a class.

211.    The data processing system of claim 206, wherein the second node comprises an interface.

212.    The data processing system of claim 206, wherein the second node comprises a package.

213.    A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;

a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that receives an indication of a second of the plurality of nodes, that determines whether the code corresponding to the first node contains a use of the second node, and when it is determined that the code corresponding to the first node contains the use of the second node, the program displays the usage of the second node by the first node; and

a processor for running the program.

214.    The data processing system of claim 213, wherein each node comprises an element.

215.    The data processing system of claim 213, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the program further displays the first node element.

216.    The data processing system of claim 215, wherein the program further displays the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

217.    The data processing system of claim 213, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the program further displays the second node element.

218.    The data processing system of claim 217, wherein the program further displays the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

219.    The data processing system of claim 213, wherein the first node comprises a class.

220.    The data processing system of claim 213, wherein the first node comprises an interface.

221.    The data processing system of claim 213, wherein the first node comprises a package.

222.    The data processing system of claim 213, wherein the second node comprises a class.

5    223.    The data processing system of claim 213, wherein the second node comprises an interface.

224.    The data processing system of claim 213, wherein the second node comprises a package.

225.    A system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the system comprising:

means for displaying a graphical representation of the dependent node;

means for displaying a graphical representation of the defining node;

5    means for receiving an indication to identify a dependency between the dependent node and the defining node; and

means for representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node in response to receiving the indication to identify the dependency.